



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/740,401	12/18/2000	David John Carew	AUS920000854US1	5846
35525	7590	03/25/2005	EXAMINER	
IBM CORP (YA) C/O YEE & ASSOCIATES PC P.O. BOX 802333 DALLAS, TX 75380			HUYNH, CONG LAC T	
			ART UNIT	PAPER NUMBER
			2178	

DATE MAILED: 03/25/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

JK

<b>Office Action Summary</b>	<b>Application No.</b> 09/740,401	<b>Applicant(s)</b> CAREW ET AL.	
	<b>Examiner</b> Cong-Lac Huynh	<b>Art Unit</b> 2178	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 10 December 2004.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1,2 and 4-34 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1,2 and 4-34 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

TL

### **DETAILED ACTION**

1. This action is responsive to communications: amendment filed 12/10/04 to the application filed on 12/18/00.
2. Claims 33-34 are added.
3. Claims 1-2, 4-34 are pending in the case. Claims 1, 9, 13, 18-19, 27, 31-32 are independent claims.
4. The rejection of claim 21 as being substantial duplicate of claim 20 has been withdrawn in view of the cancellation of claim 21.

### ***Claim Rejections - 35 USC § 102***

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-2, 4-5, 9-10, 13-23, 27-28, 31-32 remain rejected under 35 U.S.C. 102(b) as being anticipated by Brighton Beach Software (referred as Brighton Beach), Code.Spell: Source Code Spell Checker [BETA], <http://www.bbs.com.au/codespel.htm>, November 1998, Google, printouts pages 1-2.

Note: Examiner also provides the printouts from

<http://www.windevnet.com/link/subject166.htm?topic=links> and

<http://www.bbs.com.au/products.htm> to show that the date of the Code.Spell is

Art Unit: 2178

November 1998, and the publisher of the CodeSpell software is the Brighton Beach Software.

Regarding independent claim 1 and its dependent claim 4, Brighton Beach discloses:

- receiving computer source code for processing (page 1, Description: the fact that spell checking the programs of the programmers inherently shows that the computer source code in the programs is received for processing)
- responsive to a determination of a source code format, identifying displayable text within the computer source code where the text located between a set of delimiters is considered as the displayable text (page 1, Description: the fact that the text strings within the programs of the programmers can be *extracted*, saved in files and *reviewed* for logical correctness and consistency where the files are stored in ASCII format with *string delimiters such as single quotes or double-quotes* shows that the text strings between the string delimiters in the source code are identified as displayable text; further, checking strings which have single or double quotes as string delimiters shows determining the format of the strings in the source code to identify strings in the source code)
- checking the displayable text for errors (page 1, Description: spell check the displayable text strings in the programs)

Regarding claim 2, which is dependent on claim 1, Brighton Beach discloses that the computer source code is located in a resource file (page 1: the fact that the text strings

in the programs, considered as the source code are saved in the files and reviewed for logical correctness indicates that said files containing the source code are the resource files).

Regarding claim 5, which is dependent on claim 1, Brighton Beach discloses that the text is a set of literal strings (page 1).

Regarding independent claim 9, Brighton Beach discloses:

- searching source code for a first delimiter indicative of displayable text (page 1, Description: the fact that Code.Spell works with any programming language that has files stored in ASCII format and uses delimiters such as single quotes or double quotes inherently shows that to spell check the text strings in the programs, which are equivalent to the source code, Code.Spell has to search the source code in the programs for the displayable text between the two delimiters since delimiters need no spell checking; in other words, this searching includes searching for the first delimiter indicative of displayable text)
- responsive to finding the first, spell checking text after the first delimiter until a second delimiter is encountered (page 1, Description: as mentioned above, Code.Spell checks spells the text between two delimiters; in other words, spell checking text after the first delimiter until a second delimiter is performed)

Art Unit: 2178

Regarding claim 10, which is dependent on claim 9, Brighton Beach discloses that the source code is located in a file (page 1: the source code included in the programs are stored in a file).

Claims 13-17, 18 are for a data processing system of method claims 1 and 9, and are rejected under the same rationale.

Claims 19-23, 27-28 are for a system of method claims 1-5 and 9-10 respectively, and are rejected under the same rationale.

Claims 31 and 32 are for a program product of method claims 1 and 9, and are rejected under the same rationale.

7. Claims 33-34 are rejected under 35 U.S.C. 102(b) as being anticipated by Brighton Beach Software (referred as Brighton Beach), Code.Spell: Source Code Spell Checker [BETA], <http://www.bbs.com.au/codespel.htm>, November 1998, Google, printouts pages 1-2.

Regarding claim 33, which is dependent on claim 4, Beach discloses that the set of delimiters are custom delimiters determined by an author of the source code (page 1: since the source code is written by the author, the set of delimiters included in the

source are defined by the author of the source code; the delimiters, thus, are custom delimiters).

Regarding claim 34, which is dependent on claim 9, Beach discloses that the set of delimiters are custom delimiters determined by an author of the source code (page 1: since the source code is written by the author, the set of delimiters included in the source are defined by the author of the source code; the delimiters, thus, are custom delimiters).

***Claim Rejections - 35 USC § 103***

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

10. Claims 6-7, 11, 24-25, 29 remain rejected under 35 U.S.C. 103(a) as being unpatentable over Brighton Beach as applied to claims 1 and 9 above, and further in view of Kawanabe (US Pat No. 5,924,059, 7/13/99, filed 2/17/94).

Regarding claim 6, which is dependent on claim 1, Brighton Beach does not disclose explicitly that checking step includes:

- selecting a dictionary
- spell checking the displayable text using the dictionary

Kawanabe discloses spell checking includes:

- selecting a dictionary (col 1, lines 53-58, figure 7: detecting the presence/absence of a dictionary memory used for checking word spelling in dictionary ROM inherently shows that the dictionary is selected in case when the dictionary is present for checking spelling)
- spell checking the displayable text using the dictionary (abstract, figure 6, col 1, lines 49-63)

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to have combined Kawanabe into Brighton Beach since Kawanabe teaches using a dictionary for spell checking text in a document providing the advantage to incorporate into Brighton Beach for using a dictionary for spell checking the displayable text in the source code program since said text is the same as any text in a typical document.



Regarding claim 7, which is dependent on claim 1, Brighton Beach does not disclose that the dictionary is selected using a user input.

Kawanabe discloses the dictionary is selected using a user input (figure 4, col 3, lines 32-34: since the routine (detecting the presence/absence of the dictionary for selecting the dictionary) is started in response to a reset signal upon power-on of the typewriter, which is considered as an input from a user).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to have combined Kawanabe into Brighton Beach since Kawanabe discloses selecting a dictionary upon a user input providing the advantage to incorporate into Brighton Beach for offering a user a selection of a dictionary.

Regarding claim 11, which is dependent on claim 9, Brighton Beach does not disclose that the text is checked using a selected dictionary.

Kawanabe discloses that the text is checked using a selected dictionary (figure 4).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to have combined Kawanabe into Brighton Beach since Kawanabe discloses using a selected dictionary for spell checking the text providing the advantage to incorporate into Brighton Beach for spell checking the displayable text in the source code since the text in the source code is merely the typical text and so, the spell checking can also be applied on said text.

Claims 24-25, 29 are for the data processing system of method claims 6-7, 11, and are rejected under the same rationale.

11. Claims 8, 12, 26 and 30 remain rejected under 35 U.S.C. 103(a) as being unpatentable over Brighton Beach Software (referred as Brighton Beach), Code.Spell: Source Code Spell Checker [BETA], <http://www.bbs.com.au/codespel.htm>, Google, printouts pages 1-2.

Regarding claim 8, which is dependent on claim 1, Brighton Beach does not disclose locating a pointer in the source code to a resource file containing the displayable text. Instead, Brighton Beach discloses that the natural language text strings can be extracted, saved or printed and reviewed for logical correctness and consistency (page 1).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to have modified Brighton Beach to include locating a pointer in the source code to a resource file containing the displayable text since the consistency feature suggests a correspondence of displayable text in the source code and in the resource file via a pointer pointing to such a relation.

Regarding claim 12, which is dependent on claim 9, Brighton Beach does not disclose that the text is displayable when the source code is compiled and executed.

However, it would have been obvious to one of ordinary skill in the art at the time of the invention was made to have modified Brighton Beach to include the displayable feature

of the text when the source code is compiled and executed since the source code includes viewable text strings and it was well known that the source code can be compiled and executed.

Claims 26 and 30 are for the data processing system of method claims 8 and 12, and are rejected under the same rationale.

### ***Response to Arguments***

12. Applicant's arguments filed 12/10/04 have been fully considered but they are not persuasive.

Applicants argue that Beach does not disclose the feature of "responsive to a determination of a source code format, identifying displayable text within the computer source code" (Remarks, page 8).

Examiner respectfully disagrees.

Beach discloses that the text strings within the programs of the programmers can be *extracted*, saved in files and *reviewed* for logical correctness and *consistency* where the files are stored in ASCII format with *string delimiters such as single quotes or double-quotes* shows that the text strings between the string delimiters in the source code are identified as displayable text (page 1, Description). Further, checking strings which have single or double quotes as string delimiters shows determining the format of the strings in the source code to identify strings in the source code as displayable text. In

other words, identifying displayable text within the source code is performed in response to determining the format of the source code.

Applicants argue that Beach would not be able to recognize text strings in foreign languages such as Chinese or Japanese as it would try to interpret each byte separately for each character since Beach works only with ASCII format whereas the present invention can detect and recognize displayable text of various languages within the source code (Remarks, page 9).

This argument is not relevant since the argued feature is not in the claims.

Applicants argue that new claim 33 recites the feature of "wherein the set of delimiters are custom delimiters determined by an author of the source code" (Remarks, page 11). Examiner respectfully disagrees.

Since the source code is written by the author, the set of delimiters included in the source must be defined by the author of the source code. The delimiters, thus, are custom delimiters.

Applicants argue that Beach and Kawanabe do not teach "wherein the dictionary is selected using a user input" (Remarks, page 14).

Examiner respectfully disagrees.

Kawanabe discloses detecting the presence/absence of the dictionary for selecting the dictionary is started in response to a reset signal upon power-on of the typewriter (col 3,

lines 32-34). This feature indirectly shows that the dictionary is selected using a user input since a reset signal upon power-on of the typewriter is performed by a user input.

Applicants argue that Beach does not disclose "locating a pointer in the source code to a resource file containing the displayable text" since nothing in Beach states, when extracting the text strings, creates a pointer in the source code for each string, which refers to a resource file wherein the text strings are stored separately (Remarks, page 16).

Examiner agrees that Beach does not disclose "locating a pointer in the source code to a resource file containing the displayable text" explicitly.

However, Beach does teach extracting text strings in the source code for logical correctness and *consistency* (page 1).

Since the source code and the resource file, which include the text strings, are stored separately, and extracting the text strings is carried out *consistently*, there must be a correspondence between the source code and the resource file as well as a pointer in the source code pointing to the resource file so that text can be extracted and displayed properly.

### ***Conclusion***

13. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Fein et al. (US Pat No. 6,012,075, 1/4/00, filed 11/14/96).

Lee et al. (US Pat No. 6,848,080 B1, 1/25/05, filed 6/28/00, priority 11/5/99).

Robinson et al. (US Pat No. 6,801,190 B1, 10/5/04, filed 5/26/00, priority 5/27/99).

Gessner (US Pat App Pub No 2002/0032709 A1, 3/14/02, filed 9/29/98).

Schabes et al. (US Pat App Pub No 2004/0093567 A1, 5/13/04, filed 5/22/02, priority 5/26/98).

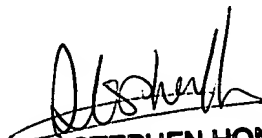
Bates et al. (US Pat App Pub No 2004/0205672 A1, 10/14/04, filed 12/29/00).

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Cong-Lac Huynh whose telephone number is 571-272-4125. The examiner can normally be reached on Mon-Fri (8:30-6:00).

Art Unit: 2178

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen Hong can be reached on 571-272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-4125.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

  
**STEPHEN HONG**  
**SUPERVISORY PATENT EXAMINER**

Clh  
3/16/05